Final Progress Report for Award FA9550-06-1-0313

Project: Trace Effect Analysis for Software Security

PI: Dr. Christian Skalka
The University of Vermont, Burlington, VT 05405

February 28, 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
28/02/2010	Final Performance Report	06/01/06-05/31/09

4. TITLE AND SUBTITLE
TRACE EFFECT ANALYSIS FOR SOFTWARE SECURITY

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-06-1-0313

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Christian Skalka

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The University of Vermont
Burlington, VT 05405

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/RSL
Suite 325
875 N. Randolph St.
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/RSL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-SR-AR-TR-10-0100

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A (Public Release).

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We developed combined run-time and compile-time analyses for enforcing trace based safety properties in higher order and Object Oriented programs, called trace effect analysis. Traces are the ordered sequence of events generated by programs. A wide variety of interesting language safety mechanisms can be expressed as trace properties, such as access control, resource usage protocols, and context sensitive flow analysis. Consequently, our analyses provide a uniform framework for automatically enforcing a large class of safety properties, which can be specialized for particular applications. Formal type theory underlies most of these analyses. We have also developed new program logics for defining access control policies. Based on temporal logics, they allow for the specification and verifiable enforcement of sophisticated security policies, and are especially useful in distributed contexts.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Christian Skalka
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	3	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(802)656-1920

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Status of Effort

This research project has been completed, and was productive and successful. We have published work establishing a rigorous theoretical foundation for our static analysis enforcing temporal properties of programs. We have extended our basic analysis to richer language models, incorporating object oriented features such as object hierarchies and dynamic dispatch. We have also performed research on authorization logics, which allow definition of highly expressive security policies. Aside from its inherent interest, this work has also led to a new research project in software security for embedded systems that is now funded under an AFOSR YIP award.

Accomplishments

Our research has made four basic contributions. First, we have shown that temporal program logics can be integrated with type analysis to enforce temporal program properties at compile time. Second, we have shown that our analyses are scalable to object oriented models. Third, we have developed new foundations for increasing practical applications of trust management systems. All of these results enhance the foundations of software security, especially software for execution in distributed environments. They lead also to our fourth contribution, which is an underpinning of new research in programming language-based security for embedded systems via type safe staged programming. Papers reporting work on this project have been published in high-profile, highly respected venues such as the Journal of Functional Programming and ACM Computing Surveys.

Personnel Supported

This grant provided Summer support for the PI during 2006, 2007, and 2008. It supported a PhD student during the 2007/2008 school year, and a postdoctoral researcher from November 1, 2007 to May 1, 2009. The grant also supported travel to conferences by the PI and funded personnel was also supported

Publications

During the grant period the PI has (co-)authored a number of papers relevant to supported research. Following are highlights.

[1] Yu David Liu, Christian Skalka, and Scott Smith. *Type-Specialized Staged Programming with Process Separation.* In Workshop on Generic Programming (WGP09), Edinburgh, Scotland, 2009.

[2] Christian Skalka. *Types and trace effects for object orientation.* Journal of Higher Order and Symbolic Computation, 21(3):239-282, 2008.

[3] Peter Chapin, Christian Skalka, and X. Sean Wang. *Authorization in Trust management: Features and foundations.* ACM Computing Surveys 40(3):1–48, 2008.

[4] Christian Skalka, Scott Smith, and David Van Horn. *Types and trace effects of higher order programs.* Journal of Functional Programming 18(2):179-249, 2008.

[5] Christian Skalka, X. Sean Wang, and Peter Chapin. *Risk management for distributed authorization.* Journal of Computer Security, 15(4):447-489, 2007.

[6] Paritosh Shroff, Christian Skalka, and Scott Smith. *The Nuggetizer: Abstracting Away Higher Orderness for Program Verification.* Proceedings of the Asian Programming Languages Symposium, November 2007.

[7] Christian Skalka. *Type safe dynamic linking for JVM access control.* Proceedings of the ACM Symposium on Principles and Practice of Declarative Programming, 2007.

[8] Christian Skalka and X. Sean Wang. *Trust but Verify: Authorization for Web Services.* Journal of Computer Systems Science and Engineering, 21(5), 2006.

[9] Jeff Polakow and Christian Skalka. Specifying distributed trust management in LolliMon. Proceedings of the ACM Workshop on Programming Languages and Analysis for Security, 2006.

Interactions: Presentations

During the grant period the PIs research has been presented at the following venues.

[1] Edinburgh University, Symposium on Data Provenance in Software, March 2009.

[2] ACM Workshop on Generic Programming, September 2009

[3] Asian Programming Languages Symposium, November 2007.

[4] McGill University Computer Science Seminar Series, June 2007.

[5] Harvard University Computer Science Seminar Series, May 2007.

[6] ACM Workshop on Programming Languages and Analysis for Security, June 2006.